관리/환경설정

Tibero7 Partition Exchange

Copyright © 2025 TmaxTibero. All Rights Reserved

TADTI188

Copyright Notice
Copyright © 2025 TIBERO Co., Ltd. All Rights Reserved.
대한민국 경기도 성남시 분당구 정자일로 45 티맥스소프트타워

Website
www.tmaxtibero.com

Restricted Rights Legend
All TIBERO Software (Tibero®) and documents are protected by copyright laws and international convention. TIBERO software and documents are made available under the terms of the TIBERO License Agreement and may only be used or copied in accordance with the terms of this agreement. No part of this document may be transmitted, copied, deployed, or reproduced in any form or by any means, electronic, mechanical, or optical, without the prior written consent of TIBERO Co., Ltd.
이 소프트웨어(Tibero®) 사용설명서의 내용과 프로그램은 저작권법과 국제 조약에 의해서 보호받고 있습니다. 사용설명서의 내용과 여기에 설명된 프로그램은 TIBERO Co., Ltd.와의 사용권 계약 하에서만 사용이 가능하며, 사용권 계약을 준수하는 경우에만 사용 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부분을 TIBERO 의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단을 사용하여 전송, 복제, 배포, 2 차적 저작물 작성 등의 행위를 하여서는 안 됩니다.

Trademarks
Tibero® is a registered trademark of TIBERO Co., Ltd. Other products, titles or services may be registered trademarks of their respective companies.
Tibero®는 TIBERO Co., Ltd.의 등록 상표입니다. 기타 모든 제품들과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용됩니다.

안내서 정보
안내서 제목: Tibero7 Partition Exchange
발행일: 2025-11-25
소프트웨어 버전: Tibero7 FS02
안내서 버전: 1.0.0

제, 개정 이력

안내서 버전	개정일자	개정 사유 및 내용	비고
1.0.0	2025.11.25	최초 제정	김찬수

목차

내용

2.3. Partition Exchange 수행 테스트

2.3.1. 테스트 사전 준비

# 파티션 테이블 생성 및 insert	SQL> CREATE TABLE TEST_PART (A NUMBER, B NUMBER, C VARCHAR(100), D DATE) NOLOGGING PARTITION BY RANGE (A) (PARTITION PART_01 VALUES LESS THAN (1), PARTITION PART_02 VALUES LESS THAN (2), PARTITION PART_03 VALUES LESS THAN (3), PARTITION PART_04 VALUES LESS THAN (4)); Table 'TEST_PART' created. /* TEST_PART Insert */ SQL> INSERT INTO TEST_PART SELECT MOD(LEVEL,4) A, LEVEL B, DBMS_RANDOM.STRING('X',100) C, TO_DATE(SYSDATE + LEVEL/24/60/60) D FROM DUAL CONNECT BY LEVEL < 100000; 99999 rows inserted. SQL> COMMIT; Commit completed.
# 파티션 인덱스 생성 (LOCAL)	SQL> CREATE INDEX TEST_PART_IDX ON TEST_PART (A,B) LOCAL NOLOGGING; Index 'TEST_PART_IDX' created.

# 교체할 일반 테이블 생성	SQL> CREATE TABLE TEST_NON TABLESPACE USR AS(SELECT * FROM TEST_PART PARTITION(PART_02) WHERE B < 50000); Table 'TEST_NON' created.
# 교체할 일반 테이블 인덱스 생성	SQL> CREATE INDEX TEST_NON_IDX ON TEST_NON (A,B) TABLESPACE USR; Index 'TEST_NON_IDX' created.
# 오브젝트 조회	SQL> SELECT SEGMENT_NAME, PARTITION_NAME, TABLESPACE_NAME FROM DBA_SEGMENTS WHERE SEGMENT_NAME LIKE 'TEST_%' ORDER BY 1, 2; SEGMENT_NAME PARTITION_NAME TABLESPACE_NAME ---------- ---------- ---------- TEST_NON USR TEST_NON_IDX USR TEST_PART PART_01 SYSTEM TEST_PART PART_02 SYSTEM TEST_PART PART_03 SYSTEM TEST_PART PART_04 SYSTEM TEST_PART_IDX PART_01 SYSTEM TEST_PART_IDX PART_02 SYSTEM TEST_PART_IDX PART_03 SYSTEM TEST_PART_IDX PART_04 SYSTEM 10 rows selected. SQL> SELECT INDEX_NAME, STATUS FROM DBA_INDEXES WHERE INDEX_NAME LIKE 'TEST_%' ORDER BY 1; INDEX_NAME STATUS ---------- -------- TEST_NON_IDX VALID TEST_PART_IDX VALID 2 rows selected.

	SQL> SELECT INDEX_NAME, PARTITION_NAME, STATUS FROM USER_IND_PARTITIONS WHERE INDEX_NAME LIKE 'TEST_%' ORDER BY 1, 2; INDEX_NAME PARTITION_NAME STATUS ------------------------------ ------------------------ -------- TEST_PART_IDX PART_01 USABLE TEST_PART_IDX PART_02 USABLE TEST_PART_IDX PART_03 USABLE TEST_PART_IDX PART_04 USABLE 4 rows selected.

2.3.2. Partition Exchange 수행 테스트 (Local Index Excluding Indexes)

※ EXCLUDING INDEXES 옵션을 사용하는 경우, 테이블 메타데이터만 교체되며 **인덱스 메타데이터는 교체되지 않습니다.** 이로 인해 **교체 대상 파티션과 연관된 인덱스는 UNUSABLE 상태가 될 수 있으므로, 해당 인덱스에 대해 별도의 REBUILD 작업이 필요**합니다.

# Partition Exchange (EXCLUDING INDEXES 적용 시)	SQL> ALTER TABLE TEST_PART EXCHANGE PARTITION PART_02 WITH TABLE TEST_NON EXCLUDING INDEXES; Table 'TEST_PART' altered.
# 오브젝트 재조회	SQL> SELECT SEGMENT_NAME, PARTITION_NAME, TABLESPACE_NAME FROM USER_SEGMENTS WHERE SEGMENT_NAME LIKE 'TEST_%' ORDER BY 1, 2; SEGMENT_NAME PARTITION_NAME TABLESPACE_NAME ------------------ ------------------------ ------------------ TEST_NON SYSTEM TEST_NON_IDX USR TEST_PART PART_01 SYSTEM TEST_PART PART_02 USR TEST_PART PART_03 SYSTEM TEST_PART PART_04 SYSTEM TEST_PART_IDX PART_01 SYSTEM TEST_PART_IDX PART_02 SYSTEM TEST_PART_IDX PART_03 SYSTEM TEST_PART_IDX PART_04 SYSTEM 10 rows selected.

	SQL> SELECT INDEX_NAME, STATUS FROM DBA_INDEXES WHERE INDEX_NAME LIKE 'TEST_%' ORDER BY 1; INDEX_NAME STATUS ------------------------------ -------- TEST_NON_IDX UNUSABLE TEST_PART_IDX UNUSABLE 2 rows selected. SQL> SELECT INDEX_NAME, PARTITION_NAME, STATUS FROM USER_IND_PARTITIONS WHERE INDEX_NAME LIKE 'TEST_%' ORDER BY 1, 2; INDEX_NAME PARTITION_NAME STATUS ------------------------------ ------------------------ -------- TEST_PART_IDX PART_01 USABLE TEST_PART_IDX PART_02 UNUSABLE TEST_PART_IDX PART_03 USABLE TEST_PART_IDX PART_04 USABLE 4 rows selected. • **파티션 PART_02의 로컬 파티션 인덱스만 UNUSABLE 되어 리빌드 필요.**

# 인덱스 리빌드	alter session set _INDEX_BUILD_USING_FULL_SCAN=Y; 	
	
	Session altered.
	
	SQL> ALTER INDEX TEST_PART_IDX REBUILD PARTITION PART_02 ONLINE;
	
	Index 'TEST_PART_IDX' altered.
	
	SQL> ALTER INDEX TEST_NON_IDX REBUILD ONLINE;
	
	Index 'TEST_NON_IDX' altered.
	
	alter session set _INDEX_BUILD_USING_FULL_SCAN=N;
	
	Session altered.

```
SQL> SELECT INDEX_NAME, STATUS
        FROM DBA_INDEXES
        WHERE INDEX_NAME LIKE 'TEST_%'
        ORDER BY 1;

INDEX_NAME                     STATUS
------------------------------ --------
TEST_NON_IDX                   VALID
TEST_PART_IDX                  VALID

2 rows selected.

SQL> SELECT INDEX_NAME, PARTITION_NAME, STATUS
        FROM USER_IND_PARTITIONS
        WHERE INDEX_NAME LIKE 'TEST_%'
        ORDER BY 1, 2;

INDEX_NAME                     PARTITION_NAME      STATUS
------------------------------ ------------------------ --------
TEST_PART_IDX                  PART_01             USABLE
TEST_PART_IDX                  PART_02             USABLE
TEST_PART_IDX                  PART_03             USABLE
TEST_PART_IDX                  PART_04             USABLE

4 rows selected.
```

2.3.3. Partition Exchange 수행 테스트 (Local Index Including Indexes)

※ INCLUDING INDEXES 옵션을 사용하는 경우, 테이블 메타데이터와 함께 교체 대상 파티션에 연관된 인덱스 메타데이터도 함께 교체됩니다. 이로 인해 **인덱스가 UNUSABLE 상태로 변경되지 않으며, 별도의 인덱스 REBUILD 작업은 필요하지 않습니다.**

 # Partition Exchange (INCLUDING INDEXES 적용 시)

```
SQL> ALTER TABLE TEST_PART EXCHANGE PARTITION PART_02 WITH TABLE TEST_NON INCLUDING INDEXES;

Table 'TEST_PART' altered.
```

 # 오브젝트 재조회

```
SQL> SELECT SEGMENT_NAME, PARTITION_NAME, TABLESPACE_NAME
       FROM USER_SEGMENTS
       WHERE SEGMENT_NAME LIKE 'TEST_%'
       ORDER BY 1, 2;

SEGMENT_NAME     PARTITION_NAME     TABLESPACE_NAME
---------------- ------------------ ----------------
TEST_NON                            SYSTEM
TEST_NON_IDX                        SYSTEM
TEST_PART        PART_01            SYSTEM
TEST_PART        PART_02            USR
TEST_PART        PART_03            SYSTEM
TEST_PART        PART_04            SYSTEM
TEST_PART_IDX    PART_01            SYSTEM
TEST_PART_IDX    PART_02            USR
TEST_PART_IDX    PART_03            SYSTEM
TEST_PART_IDX    PART_04            SYSTEM

10 rows selected.
```

• TEST_NON과 PART_02의 데이터 및 인덱스 변경 확인

```
SQL> SELECT INDEX_NAME, STATUS
       FROM DBA_INDEXES
       WHERE INDEX_NAME LIKE 'TEST_%'
       ORDER BY 1;

INDEX_NAME                     STATUS
------------------------------ --------
TEST_NON_IDX                   VALID
TEST_PART_IDX                  VALID

2 rows selected.

SQL> SELECT INDEX_NAME, PARTITION_NAME, STATUS
```

```
FROM USER_IND_PARTITIONS
WHERE INDEX_NAME LIKE 'TEST_%'
ORDER BY 1, 2;

INDEX_NAME                     PARTITION_NAME          STATUS
------------------------------ ----------------------- --------
TEST_PART_IDX                  PART_01                 USABLE
TEST_PART_IDX                  PART_02                 USABLE
TEST_PART_IDX                  PART_03                 USABLE
TEST_PART_IDX                  PART_04                 USABLE

4 rows selected.
```

2.3.4 테스트 사전 준비

# 파티션 테이블 생성 및 insert	SQL> CREATE TABLE TEST_PART (A NUMBER, B NUMBER, C VARCHAR(100), D DATE) NOLOGGING PARTITION BY RANGE (A) (PARTITION PART_01 VALUES LESS THAN (1), PARTITION PART_02 VALUES LESS THAN (2), PARTITION PART_03 VALUES LESS THAN (3), PARTITION PART_04 VALUES LESS THAN (4)); Table 'TEST_PART' created. /* TEST_PART Insert */ SQL> INSERT INTO TEST_PART SELECT MOD(LEVEL,4) A, LEVEL B, DBMS_RANDOM.STRING('X',100) C, TO_DATE(SYSDATE + LEVEL/24/60/60) D FROM DUAL CONNECT BY LEVEL < 100000; 99999 rows inserted. SQL> COMMIT; Commit completed.
# 파티션 인덱스 생성 (GLOBAL)	SQL> CREATE INDEX TEST_PART_GLOBAL_IDX ON TEST_PART (A, B) GLOBAL PARTITION BY RANGE (A) (PARTITION P1 VALUES LESS THAN (1), PARTITION P2 VALUES LESS THAN (2), PARTITION P3 VALUES LESS THAN (3), PARTITION P4 VALUES LESS THAN (4), PARTITION P5 VALUES LESS THAN (MAXVALUE)); Index 'TEST_PART_GLOBAL_IDX' created.
# 교체할 일반 테이블 생성	1.2절의 교체할 일반 테이블 및 데이터 Insert 구문과 동일합니다.
# 교체할 일반 테이블 인덱스 생성	SQL> CREATE INDEX TEST_NON_GLOBAL_IDX ON TEST_NON (A,B) TABLESPACE USR;

	Index 'TEST_NON_GLOBAL_IDX' created.
# 오브젝트 조회	SQL> SELECT SEGMENT_NAME, PARTITION_NAME, TABLESPACE_NAME FROM USER_SEGMENTS WHERE SEGMENT_NAME LIKE 'TEST_%' ORDER BY 1, 2; SEGMENT_NAME PARTITION_NAME TABLESPACE_NAME ---------- ---------- ---------- TEST_NON_GLOBAL USR TEST_NON_GLOBAL_IDX USR TEST_PART_GLOBAL PART_01 SYSTEM TEST_PART_GLOBAL PART_02 SYSTEM TEST_PART_GLOBAL PART_03 SYSTEM TEST_PART_GLOBAL PART_04 SYSTEM TEST_PART_GLOBAL_IDX P1 SYSTEM TEST_PART_GLOBAL_IDX P2 SYSTEM TEST_PART_GLOBAL_IDX P3 SYSTEM TEST_PART_GLOBAL_IDX P4 SYSTEM TEST_PART_GLOBAL_IDX P5 SYSTEM 11 rows selected. SQL> SELECT INDEX_NAME, STATUS FROM USER_INDEXES WHERE INDEX_NAME LIKE '%TEST_%' ORDER BY 1; INDEX_NAME STATUS ---------- -------- TEST_NON_IDX VALID TEST_PART_GLOBAL_IDX VALID 2 rows selected. SQL> SELECT INDEX_NAME, PARTITION_NAME, STATUS FROM USER_IND_PARTITIONS WHERE INDEX_NAME LIKE '%TEST_%' ORDER BY 1, 2; INDEX_NAME PARTITION_NAME STATUS ---------- ---------- -------- TEST_PART_GLOBAL_IDX P1 USABLE TEST_PART_GLOBAL_IDX P2 USABLE TEST_PART_GLOBAL_IDX P3 USABLE

	TEST_PART_GLOBAL_IDX P4 USABLE TEST_PART_GLOBAL_IDX P5 USABLE 5 rows selected.

2.3.5. Partition Exchange 수행 시나리오 및 테스트 (Global Index 인 경우 Excluding Indexes)

# Partition Exchange (EXCLUDING INDEXES 적용 시)	SQL> ALTER TABLE TEST_PART EXCHANGE PARTITION PART_02 WITH TABLE TEST_NON EXCLUDING INDEXES; Table 'TEST_PART' altered.
# 오브젝트 재조회	SQL> SELECT SEGMENT_NAME, PARTITION_NAME, TABLESPACE_NAME FROM USER_SEGMENTS WHERE SEGMENT_NAME LIKE 'TEST_%' ORDER BY 1, 2; SEGMENT_NAME PARTITION_NAME TABLESPACE_NAME ---------- ---------- ---------- TEST_NON SYSTEM TEST_NON_IDX USR TEST_PART PART_01 SYSTEM TEST_PART PART_02 USR TEST_PART PART_03 SYSTEM TEST_PART PART_04 SYSTEM TEST_PART_GLOBAL_IDX P1 SYSTEM TEST_PART_GLOBAL_IDX P2 SYSTEM TEST_PART_GLOBAL_IDX P3 SYSTEM TEST_PART_GLOBAL_IDX P4 SYSTEM TEST_PART_GLOBAL_IDX P5 SYSTEM 11 rows selected. SQL> SELECT INDEX_NAME, STATUS FROM USER_INDEXES WHERE INDEX_NAME LIKE '%TEST_%' ORDER BY 1; INDEX_NAME STATUS ---------- -------- TEST_NON_IDX UNUSABLE TEST_PART_GLOBAL_IDX UNUSABLE 2 rows selected.

	SQL> SELECT INDEX_NAME, PARTITION_NAME, STATUS FROM USER_IND_PARTITIONS WHERE INDEX_NAME LIKE '%TEST_%' ORDER BY 1, 2; INDEX_NAME PARTITION_NAME STATUS ------------------------------ ------------------------ -------- TEST_PART_GLOBAL_IDX P1 UNUSABLE TEST_PART_GLOBAL_IDX P2 UNUSABLE TEST_PART_GLOBAL_IDX P3 UNUSABLE TEST_PART_GLOBAL_IDX P4 UNUSABLE TEST_PART_GLOBAL_IDX P5 UNUSABLE 5 rows selected.
# 인덱스 리빌드	ALTER INDEX TEST_PART_GLOBAL_IDX REBUILD PARTITION P1 ONLINE; .. ALTER INDEX TEST_PART_GLOBAL_IDX REBUILD PARTITION P5 ONLINE; SQL> ALTER INDEX TEST_NON_IDX REBUILD ONLINE; Index 'TEST_NON_IDX' altered. INDEX_NAME STATUS ------------------------------ -------- TEST_NON_IDX VALID TEST_PART_GLOBAL_IDX VALID 2 rows selected. INDEX_NAME PARTITION_NAME STATUS ------------------------------ ------------------------ -------- TEST_PART_GLOBAL_IDX P1 USABLE TEST_PART_GLOBAL_IDX P2 USABLE TEST_PART_GLOBAL_IDX P3 USABLE TEST_PART_GLOBAL_IDX P4 USABLE TEST_PART_GLOBAL_IDX P5 USABLE

2.3.6. Partition Exchange 수행 시나리오 및 테스트 (Global Index 인 경우 Including Indexes)

※ Global Index의 경우 전체 파티션을 대상으로 구성되기 때문에 특정 파티션만 교체하는 Partition Exchange 시 INCLUDING INDEXES 옵션을 사용할 수 없습니다.

 # Partition Exchange (INCLUDING INDEXES 적용 시)

```
SQL> ALTER TABLE TEST_PART EXCHANGE PARTITION PART_02 WITH TABLE TEST_NON INCLUDING INDEXES;
TBR-7281: Allowed only for a local partitioned index.

SQL> !tberr 7281
/*
 * err:   -7281
 * name:  ERROR_DDL_ALLOWED_LOCAL_INDEX_ONLY
 * desc:   Allowed only for a local partitioned index.
 * cause:  Can be executed only for a local partitioned index.
 * action: Unable to execute this command.;
 */

SQL> ALTER TABLE TEST_PART EXCHANGE PARTITION PART_02 WITH TABLE TEST_NON EXCLUDING INDEXES;

Table 'TEST_PART' altered.
```

 # 오브젝트 재조회

```
SQL> SELECT SEGMENT_NAME, PARTITION_NAME, TABLESPACE_NAME
        FROM USER_SEGMENTS
        WHERE SEGMENT_NAME LIKE 'TEST_%'
        ORDER BY 1, 2;

SEGMENT_NAME          PARTITION_NAME     TABLESPACE_NAME
--------------------- -------------------------- --------------------
TEST_NON                                 SYSTEM
TEST_NON_IDX                             USR
TEST_PART             PART_01            SYSTEM
TEST_PART             PART_02            USR
TEST_PART             PART_03            SYSTEM
TEST_PART             PART_04            SYSTEM
TEST_PART_GLOBAL_IDX  P1                  SYSTEM
TEST_PART_GLOBAL_IDX  P2                  SYSTEM
TEST_PART_GLOBAL_IDX  P3                  SYSTEM
TEST_PART_GLOBAL_IDX  P4                  SYSTEM
TEST_PART_GLOBAL_IDX  P5                  SYSTEM

SQL> SELECT INDEX_NAME, STATUS
        FROM USER_INDEXES
        WHERE INDEX_NAME LIKE '%TEST_%'
        ORDER BY 1;
```

	INDEX_NAME STATUS ------------------------------ -------- TEST_NON_IDX UNUSABLE TEST_PART_GLOBAL_IDX UNUSABLE 2 rows selected. SQL> SELECT INDEX_NAME, PARTITION_NAME, STATUS FROM USER_IND_PARTITIONS WHERE INDEX_NAME LIKE '%TEST_%' ORDER BY 1, 2; INDEX_NAME PARTITION_NAME STATUS ------------------------------ ------------------------ -------- TEST_PART_GLOBAL_IDX P1 UNUSABLE TEST_PART_GLOBAL_IDX P2 UNUSABLE TEST_PART_GLOBAL_IDX P3 UNUSABLE TEST_PART_GLOBAL_IDX P4 UNUSABLE TEST_PART_GLOBAL_IDX P5 UNUSABLE 5 rows selected.
# 인덱스 리빌드	ALTER INDEX TEST_PART_GLOBAL_IDX REBUILD PARTITION P1 ONLINE; .. ALTER INDEX TEST_PART_GLOBAL_IDX REBUILD PARTITION P5 ONLINE; SQL> ALTER INDEX TEST_NON_IDX REBUILD ONLINE; Index 'TEST_NON_IDX' altered. INDEX_NAME STATUS ------------------------------ -------- TEST_NON_IDX VALID TEST_PART_GLOBAL_IDX VALID 2 rows selected. INDEX_NAME PARTITION_NAME STATUS ------------------------------ ------------------------ -------- TEST_PART_GLOBAL_IDX P1 USABLE TEST_PART_GLOBAL_IDX P2 USABLE TEST_PART_GLOBAL_IDX P3 USABLE TEST_PART_GLOBAL_IDX P4 USABLE TEST_PART_GLOBAL_IDX P5 USABLE

	5 rows selected.

2.3.7. Partition Exchange 수행 테스트 (Partition key 위배 시 WITH VALIDATION)

※ VALIDATION 옵션은 Local Index 유무와 무관하게 사용 가능하나, Partition과 교체 대상 테이블 간의 컬럼·제약조건·인덱스 구조가 동일해야 하며, 구조 불일치 시 유효성 검사 단계에서 오류가 발생합니다.

# TEST_NON 테이블에 1이 아닌 값 Insert	**• TEST_NON A 컬럼의 값은 1이며, part_02의 Partition key 값도 1이다.** **• TEST_NON 테이블에 1이 넘는 값을 넣고, Partition exchange 수행한다.** SQL> INSERT INTO TEST_NON SELECT CEIL(DBMS_RANDOM.VALUE(1,10)) A, LEVEL B, DBMS_RANDOM.STRING('X',100) C, TO_DATE(SYSDATE + LEVEL/24/60/60) D FROM DUAL CONNECT BY LEVEL <= 1000; 1000 rows inserted. SQL> COMMIT; Commit completed.
# Partition Exchange 수행 (WITH VALIDATION)	SQL> ALTER TABLE TEST_PART EXCHANGE PARTITION PART_02 WITH TABLE TEST_NON WITH VALIDATION; TBR-7287: Some rows in the table are invalid for the specified partition. SQL> !tberr 7287 /* * err: -7287 * name: ERROR_DDL_ROWS_DONT_QUALIFY_FOR_PARTITION * desc: Some rows in the table are invalid for the specifiedpartition. * cause: Some rows cannot be inserted in the specifiedpartition. * action: Remove the invalid rows and try again.; */ **• Partition exchange 수행 시, TEST_NOE 테이블의 몇몇 데이터들이 partition key 값이 아닌 데이터가 존재한다는 뜻.** *** 해결 방안** **• Partition key를 위배하는 데이터를 찾아 delete 수행 필요** **• WITHOUT VALIDATION 옵션을 주어 강제로 Partition Exchange 수행 필요**

2.3.8. Partition Exchange 수행 테스트 (Partition key 위배 시 WITHOUT VALIDATION)

※ WITHOUT VALIDATION 옵션 사용 시, 파티션 키 유효성 검사를 생략하므로, 파티션 키를 위배한 데이터가 파티션에 포함될 수 있으며, 이후 DML 수행 시 데이터 정합성 문제 및 오류 및 DDL 수행에 대한 오류가 발생할 수 있음에 유의해야 합니다.

 # Partition Exchange 수행 (WITHOUT VALIDATION)

```
SQL> ALTER TABLE TEST_PART EXCHANGE PARTITION PART_02 WITH
TABLE TEST_NON WITHOUT VALIDATION;

Table 'TEST_PART' altered.
```

 # 오브젝트 조회

```
SQL> SELECT SEGMENT_NAME, PARTITION_NAME, TABLESPACE_NAME
        FROM USER_SEGMENTS
        WHERE SEGMENT_NAME LIKE 'TEST_%'
        ORDER BY 1, 2;

SEGMENT_NAME    PARTITION_NAME      TABLESPACE_NAME
------------------- ------------------------ --------------------
TEST_NON                                     USR
TEST_NON_IDX                                 USR
TEST_PART           PART_01                  SYSTEM
TEST_PART           PART_02                  SYSTEM
TEST_PART           PART_03                  SYSTEM
TEST_PART           PART_04                  SYSTEM
TEST_PART_IDX       PART_01                  SYSTEM
TEST_PART_IDX       PART_02                  SYSTEM
TEST_PART_IDX       PART_03                  SYSTEM
TEST_PART_IDX       PART_04                  SYSTEM

10 rows selected.

SQL> SELECT INDEX_NAME, STATUS
       FROM USER_INDEXES
       WHERE INDEX_NAME LIKE 'TEST_%'
       ORDER BY 1;

INDEX_NAME                  STATUS
------------------------------ --------
TEST_NON_IDX                UNUSABLE
TEST_PART_IDX               UNUSABLE

2 rows selected.

SQL> SELECT INDEX_NAME, PARTITION_NAME, STATUS
```

	FROM USER_IND_PARTITIONS WHERE INDEX_NAME LIKE 'TEST_%' ORDER BY 1, 2; INDEX_NAME PARTITION_NAME STATUS ------------------------------ ------------------------ -------- TEST_PART_IDX PART_01 USABLE TEST_PART_IDX PART_02 UNUSABLE TEST_PART_IDX PART_03 USABLE TEST_PART_IDX PART_04 USABLE 4 rows selected.
# 1) 인덱스 리빌드	SQL> alter session set _INDEX_BUILD_USING_FULL_SCAN=Y; Session altered. SQL> ALTER INDEX TEST_PART_IDX REBUILD PARTITION PART_02; Index 'TEST_PART_IDX' altered. SQL> ALTER INDEX TEST_NON_IDX REBUILD ; Index 'TEST_NON_IDX' altered. SQL> alter session set _INDEX_BUILD_USING_FULL_SCAN=N; Session altered.
# 위배 데이터 조회	SQL> SELECT COUNT(*) FROM TEST_PART PARTITION(PART_02) WHERE A <> 1; COUNT(*) ---------- 1000 1 row selected.